IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/557,738	Confirmation No.	9935
Appellants	:	Kevin B. Gjerstad, *et al.*		
Filed	:	April 25, 2000		
Group Art Unit	:	2176		
Examiner	:	William L. Bashore		
Title	:	COMMON TEXT FRAMEWORK		
Atty. Docket No.	:	MFCP.87507		
Customer No.	:	45809		

VIA ELECTRONIC FILING SYSTEM – August 31, 2006

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO NOTICE OF NON-COMPLIANT APPEAL BRIEF

Dear Sir:

This is in response to the Notice of Non-Complaint Appeal Brief dated August 23, 2006. The Commissioner is hereby authorized to charge any additional fee that may be due, or credit any overpayment, to Deposit Account No. 19-2112.

Table of Contents

I.   **REAL PARTY IN INTEREST**

The real party in interest is MICROSOFT CORPORATION, duly organized and existing under the laws of the State of WASHINGTON, United States of America. The mailing address for purposes of this Appeal is One Microsoft Way, Redmond, WA 98052.

II.   RELATED APPEALS AND INTERFERENCES

None.

III.   STATUS OF CLAIMS

Claims 1-5 and 20-26 are pending, and the rejection of each of those claims is being appealed. Claims 6-19 are cancelled.

IV.   STATUS OF AMENDMENTS

An amendment filed on 2/10/06 subsequent to the final Office Action dated 12/21/2005

was denied entry.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The instant Application includes independent claims, 1, 20, and 23.

Claim 1

Claim 1 is directed to a machine-readable medium having instructions stored thereon for execution by a processor to implement a computer program providing a common text framework **202** through which applications **204** and handlers **208** for input devices **210** can interact with a document **206**. See, i.e., Appellant's specification page 11, lines 4-20. The common text framework **202** accesses and modifies documents **206** via a text store interface. See, i.e., Appellant's specification pages 12, line 19-page 13, line 3. The text store interface includes a text stream and dynamic text interface **400** and **500**. In the text stream interface **400** the abstraction of the document appears as an array having a position within the document represented as an offset from a beginning of the array. In the dynamic text interface **500** the abstraction of the document is such that a position within the document is represented as a floating anchor to a node. See, i.e., Appellant's specification page 13, lines 5-20. The common text framework **202** communicates data received from a plurality of input devices **210** to the document **206** or **602** by utilizing the abstractions and a text input processor interface **600**. See, i.e., Appellant's specification page 14, lines 21-23. In other words, the text input processor interface **600** permits the handlers **208** to insert additional text into the document **602**. Id. Moreover, the text input processor interface **600** implements a text processor input method that attaches properties **700** to ranges within the document **602**. See Appellant's specification page 16, line 22-page 17, line 5. The text processor input method utilizes the properties **700** to preserve originally entered data received by the handlers **208**. Accordingly, the preserved data may be utilized for text correction. See, i.e., Appellant's specification page 17, lines 5-15.

Claim 20

Claim 20 is directed to a machine-readable medium having instructions stored thereon for execution by a processor to implement a computer program providing a common text framework **202** through which applications **204** and handlers **208** for input devices **210** can interact with a document **602** via a text input processor interface **600** that permits the handlers **208** associated with the input devices **210** to access an abstraction of the document **602** and to utilize the abstraction to insert additional text into the document **602**. See , i.e., Appellant's specification page 14, lines 21-23. The text input processor interface **600** provides a range object **612** in which a range within the document is specified as two positions within the abstraction of the document **602**, such that the handler inserts the additional text into the document **602** and accesses the abstraction of the document **602** at the range specified by the range object **612**. See, i.e., Appellant's specification page 15, lines 3-7. Additionally, the text input processor interface **600** includes a first method by which insertion is accomplished, a second method by which access is accomplished, and a third method by which the handler **208** is able to attach a property **700** to the document **602** at the range specified by the range object **612**, the property **700** providing access to original data used for insertion of text within the range. See, i.e., Appellant's specification page 16, lines 10-19 and page 17, lines 1-10.

Claim 23

Claim 23 is directed to a method to preserve original raw data in a common text framework **202** that provides interaction between an application **204** and a plurality of input devices **210**. See, i.e., Appellant's specification page 4, lines 10-15. The original raw data is received from at least one of the plurality of input devices **210**. See, i.e., Id. In turn, the original raw data is stored in a property **700** that is attachable to a document **602**. See, i.e., Appellant's specification page 17, lines 4-10. A range object **612, 614** is utilized to update the

document **602** by specifying a context and appropriate insertion point for text associated with the original raw data. See, i.e., Appellant's specification page 17, lines 5-10. Thereafter, the property **700** having the original raw data is attached to the range specified within the document **602**. See, i.e., Appellant's specification page 17, lines 10-13.

VI.   GROUNDS OF REJECTIONS TO BE REVIEWED ON APPEAL

A)   Whether claims 1-5 and 22 are unpatentable under 35 U.S.C. §103(a) over U.S.

Patent No. 5,946,499 to Saunders (hereinafter "Saunders") in view of U.S. Patent No.

6,466,240 to Maslov (hereinafter "Maslov") and U.S. Patent No. 5,109,439 to Froessl

(hereinafter "Froessl").

B)   Whether claims 20, 21 and 23-26 are unpatentable under 35 U.S.C. §103(a) over

Saunders in view of Froessl.

VII.   ARGUMENT

A *prima facie* case of obviousness requires, among other things, 1) prior-art references that teach or suggest all the claimed limitations; 2) a suggestion or motivation to modify the prior-art references or to combine the prior-art references; and 3) a reasonable expectation of success. *See* MPEP § 2143 . Moreover, the teaching or suggestion, and the reasonable expectation of success must be found in the prior art and not based on Appellant's disclosure. *See* MPEP §§ 706.02(j), 2142 and 2143. Appellant respectfully submits that a *prima facie* case of obviousness has not been established.

A) Rejection under 35 U.S.C. § 103(a) over Saunders in view of Maslov and Froessl.
(i) Claims 1-5 and 22

The Office Action fails to provide one or more references that reasonably teach or suggest the claimed text store interface having at least a text stream interface, a dynamic text interface and a text input method. The cited references, Saunders, Maslov and Froessl, fail to teach or suggest, among other things, text store and text input processor interfaces as claimed in claims 1-4. Accordingly, the Office has failed to meet its burden of establishing a *prima facie* case because all the claimed requirements are not suggested or taught by the cited references.

"All words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson, 424 F.2d 1382, 1385 (CCPA 1970). Generally, the Office Action does not seem to read all words of the claimed requirements together. Instead, the Office Action provides a piecemeal analysis of the claimed embodiment.

For instance, the Office Action states Saunders discloses the text processor input method, but then later states that Saunders fails to disclose a property that preserves originally entered data. The claimed requirement when read together expressly indicates a "text

processor input method for attaching a property to the document in at least one position in the document, wherein the property preserves originally entered data." This claimed requirement expressly indicates the property that preserves original data is provided by the text input processor method. Accordingly, if Saunders does not disclose the claimed property that preserves originally entered data then it follows that Saunders does not disclose the text input processor method for attaching the property that preserves originally entered data. Moreover, Saunders column 4, lines 15-18, expressly discloses that the data structure that provides access to the document is destroyed. Therefore, Saunders teaches away from a property that preserves the original data.

Saunders, Maslov and Froessl fail to disclose a text store interface having the claimed "text stream interface."

The claims requires, among other things, "a text store interface to permit an application having a document of primarily text to expose the document as an abstraction, the text store interface comprising, a text stream interface in which the abstraction of the document appears as an array, a position within the document represented as an offset from a beginning of the array, a dynamic text interface . . . and a text processor input method." That is, the metes and bounds of the claimed embodiment requires the text store interface to expose a document as an abstraction. The abstraction may be accessed via the text stream interface and dynamic text interfaces provided by the text store interface. The text stream interface provides an abstraction that represents the document as an array and offset, the offset representing a position in the array. The dynamic text interface provides an abstraction that represents the position as a floating anchor to a node. Moreover, the text store interface includes a text input method for attaching a property, which preserves originally entered data, to the document.

The Office Action seems to concede Saunders fails to disclose or suggest the claimed dynamic text interface and the text input method. However, the Office Action contends Saunders discloses the claimed "text store interface." Appellant respectfully disagrees. The claimed "text store interface" requires, among other things, a text stream interface, a dynamic text interface and a text input processor method. Because the Office action concedes Saunders does not disclose the dynamic text interface and the text input processor method, Saunders cannot disclose the claimed "text store interface."

Nonetheless, the Office Action cites Saunders, Fig.1, 4a, 4b, column 3, lines 53-56 and column 7, lines 18-32, to support its contention that Saunders discloses a document being exposed as an abstraction, such as an array having positions of the document represented as offsets from a beginning of the array. Although Saunders, column 3, lines 56, generally discloses an array data structure, Saunders fails to disclose or suggest an array having an offset from the beginning of the array, the offset representing a position within the document. In column 6, lines 20-26, Saunders discloses the storage model is a flat stream of contiguous bytes ranging from 0 to n, n being the total number of bytes in the text content. The disclosed contiguous stream is not array associated with an offset, so it is not an array having an offset address.

Saunders, column 7, lines 20-30 discloses position-to-offset and offset-to-position operations that allow text services to be completely independent from the manner in which the application program lays out the text and rather allows the mapping of changes to the device dependent location of the text. The operations map a device position to an offset within the text of a reserved portion associated with a text object. As discusses below, while Saunders

does disclose an "offset," the claimed offset is a position within the document which is different from Saunders' "offset."

By way of context, Saunders, column 6, lines 30-31, expressly discloses a region is reserved before any operation is performed on the region by the text service. Furthermore, Saunders, column 6, lines 40-41, discloses the regions relate to portions of a document like paragraphs, words, highlighted sections, etc. Additionally, Saunders, column 6, lines 44-47, discloses the specified region is then exchanged as a text object, which is an opaque data structure. At best, Saunders may utilize the opaque data structure, such as an array to provide an offset that represents a position within a reserved region of text, such as a paragraph, and the position-to-offset and offset-to-position operations are utilized to map a device to a location in the reserved region. See column 7, lines 1-2. Unlike, Saunders the claimed embodiment utilizes an offset to the beginning of an array to represent a position within the document not a position within a reserved portion or segment of the document as disclosed by Saunders in column 7, lines 1-30. Accordingly, Saunders is factually deficient with respect to the claimed "text store interface" that exposes a document as an abstraction via a text stream interface that provides the abstraction as an array that utilizes an offset from the beginning of the array to represent a position in the document.

Even if Saunders discloses the claimed text stream interface, the Office Action does not provide a reference that discloses or suggests a text store interface having, among other things, a text stream interface and a dynamic text interface in which the abstraction of the document is such that a position within the document is represented as a floating anchor to a node.

The Office Action concedes that Saunders fails to disclose the dynamic text interface and relies on Maslov to teach or suggest the claimed dynamic text interface. The Office Action

cites Maslov, column 2, line 62-column 3, line 17 and column 3, line 26-44, to support its contention that Maslov discloses or suggests a dynamic text interface, which provides an abstraction that represent a position in document as a floating anchor to a node. While Maslov discloses a tree structure having nodes, Maslov is directed to synchronizing a first display having a tree and a second display having structured text. Maslov, column 3, lines 45-55, discloses various visual operations that may be performed on the tree, but fails to indicate the relationships between the node and positions within the document. In the claimed embodiment, the floating anchor nodes exposes a position within a document, unlike Maslov, which synchronizes content on two windows. Maslov does not mention the word position anywhere in the disclosure and fails to provide a floating anchor that tracks positions within the document. Rather Maslov, column 4, lines 20-40 discloses selecting a node in the first window and highlighting the corresponding programming text. Accordingly, Saunders and Maslov are factually deficient with respect to the claimed "text store interface" having a dynamic text interface that exposes a position in the document as a floating anchor to a node.

Even if the combination of Maslov and Saunders is proper, the Office Action fails to disclose or suggest, among other things, a text store interface having a text stream interface, a dynamic text interface and a text processor input method "for attaching a property to the document in at least one position in the document, wherein the property preserves originally entered data."

The Office Action cites Saunders for disclosing a text processor input method that attaches a property in at least one position in the document, but concedes Saunders fails to disclose the property preserves originally entered data in order to facilitate text correction. Despite the Office Action's characterization of the reservation identifier, the reservation

identifier as disclosed by Saunders cannot be preserved in a position within the document because Saunders, column 4, lines 15-18, expressly discloses the data structure that represents the document is destroyed after operations are completed on the document. The Office Action attempts to equate the reservation identifier disclosed by Saunders to the claimed property. This is improper because the reservation identifier indicates whether a range of text is being accessed by another text service. Additionally, the reservation identifier differs from the property because the reservation identifier is not attached to a position in the document. At best, Saunders, column 4, lines 15-30 and column 6, lines 30-40 and 55-60, discloses that the reservation identifier is associated with the TSM context data structure which is attached to the TSM document data structure that is destroyed after completing an operation on the reserved portion of the text. Nothing in Saunders discloses attaching the property to a position in the document. Rather Saunders, column 6, lines 65-67, discloses the reservation identifier is a Boolean identifier that indicates availability of specified region. The Office Action attempts to modify the reservation identifier, a Boolean identifier to be similar to the claimed property that preserves original data by citing Froessl, column 7, lines 65-column 8, lines 5. In operation, Froessl, column 3, lines 60-65, column 5, lines 15-30, column 8, lines 1-5 disclose scanning a document and storing an image of the document in temporary memory and converting either selected portions or the entire document to ASCII code based on optical character recognition operations and storing the ASCII in permanent storage. Froessl further discloses an identifier is utilized to associate the ASCII and the image such that when a flag is triggered a human reader may perform corrections based on the image stored in temporary memory. Although Froessl discloses preserving original data, Froessl fails to attach a property that preserves the original data in the document. Rather, Froessl associates image data and the ASCII via the

identifier that correlates the ASCII and image data that is stored in temporary memory. Unlike Froessl, the claimed embodiment attaches the property to a position in the document and the property preserves the original data to facilitate correction. Accordingly, Saunders, Maslov and Froessl are factually deficient with respect to the claimed "text store interface" having a text processor input method that attaches a property that preserves originally entered data in at least one position in the document.

Therefore, the Office Action fails to provide one or more references that reasonably teach or suggest the claimed text store interface.

B) Rejection under 35 U.S.C. § 103(a) over Saunders in view of Froessl.

(i) Claims 20-21

The Office Action fails to provide one or more references that reasonably teach or suggest the claimed text input processor interface having at least a range object and a third method by which a handler is able to attach a property to the document at the range specified by the range object, the property providing access to original data. The cited references, Saunders and Froessl, fail to teach or suggest, among other things, the range object and the third method. Accordingly, the Office has failed to meet its burden of establishing a *prima facie* case because all the claimed requirements are not suggested or taught by the cited references.

The Office Action relies on Saunders and Froessl to support a conclusion that the claimed text input processor interface is unpatentable. Applicants disagree. The claimed text input processor interface requires, among other things, a range object and third method. Saunders and Froessl individually and in combination fail to provide a text input processor interface because neither reference discloses the claimed range object or third method.

The Office action cites Saunders, column 1, line 55-column 2, line 29, column 4, line 59-column 5, line 9 and column 6, line 39-column 7, line 44, for disclosing a text input processor interface that permits a handler of an input device to access an abstraction of a document and to insert "additional text" into the document. The cited portions of Saunders detail a system having multiple devices that provide input to handlers that process the input to provide appropriate textual outputs. The inputs may be utilized for example to reserve a portion of text and perform a check that provides replacement text for the portion. Unlike, Saunders the claimed text input processor interface inserts "additional text," which is very different from merely fixing content associated with text.

Even if Saunders discloses the claimed insertion of additional text, Saunders still fails to disclose the range object that specifies a range within in the document as two positions within the abstraction of the documents, such that the handler accesses the abstraction and inserts the additional text at the range specified.

The Office action cites Saunders column 6, line 39-column 7, line 44 to support a conclusion that the Saunders discloses the range object. Appellant respectfully disagrees. In column 6, lines 30-40, each portion of text is reserved by associating an owner, an unique identifier, and a range with the portion. The range is the set of bytes for a portion of the flat stream being reserved. In column 6, lines 45-55, Saunders further clarifies that "all" text exchanges are done with a text object, where the application program creates the text object with the text itself and regions of the text. In column 6, line 63, Saunders further details that once the region is obtained, the range for the region is identified. Nothing in cited portion mentions the word position. The claimed range object utilizes two positions in the abstraction of the document to allow the handler to insert additional text. At best, Saunders discloses a set

of bytes that are processed by an application to create a text object, where operations are performed on the text object once the range is known. Saunders fails to discloses a handler that utilizes a text input processor having a range object with two positions in the document that define a range where text is accessed and additional text is inserted.

Moreover, even if Saunders discloses the claimed text input processor having the range object, Saunders still fails to disclose the claimed third method by which the handler is able to attach a property to the document at the range specified by the range object, the property providing access to original data used for insertion of text within the range.

The Office action is unclear as to whether Saunders discloses the claimed third method. The Office action seems to indicate that Saunders teaches the claimed third method, but does not teach the property that provides access to original data. The claimed requirement expressly states "a third method by which the handler is able to attach a property to the document at the range specified by the range object, the property providing access to original data used for insertion of text within the range." Reading all the words of the claimed requirement together leads one to reasonable conclude if Saunders does not disclose the claimed property it follows that Saunders does not disclose the claimed third method that attaches the claimed property. Thus, the Office Action seems to concede that Saunders does not disclose or suggest the claimed third method.

The proposed combination of Saunders and Froessl to obtain the claimed property teaches away from the disclosed purpose of the reservation identifier and the functions associated with the data structures used by Saunders.

Saunders column 4, lines 15-18, discloses destruction of the document data structure, which seems to teach away from the claimed property that preserves originally entered data.

Saunders destroys the data structure and does not attach the claimed property to the text storage model. Saunders cannot be successfully combined with a teaching of preserving the original data in a property because Saunders teaches away from preserving originally entered data via a property that is attached in the document. Moreover, Saunders, column 6, lines 30-35, discloses the reservation identifier is not associated with the storage model, rather it is associated with the TSM context or document, a separate structure, which is discarded after the operations are completed. Also, the reservation identifier does not preserve the original data, rather the reservation identifier is an access flag that notifies the text services when a specified region is available. Because Saunders expressly discloses destruction of the data structures and there is no teaching or suggestion of preserving the original data, there is no modification to Saunders that renders the claimed property a viable modification. See MPEP § 2141.02 (prior art must be considered in its entirety, including disclosures that teach away from the claims) and MPEP § 2143.01 (proposed modification cannot render the prior art unsatisfactory for its intended purpose or change the principle of operation of a reference).

Nevertheless, even if Saunders can be modified as suggested, Froessl does not provide the claimed property that is attached to range in the document, the property providing access to the original data used for insertion.

The Office Action cites Froessl column 7, line 63-col 8, line 5 for teaching attaching a property to a range in a document, where the property preserves originally entered data. As discussed above, while Froessl discloses preserving original data, Froessl fails to attach a property that preserves the original data in the range of the document. Rather Froessl column 8, line 20-25 associates image data and the ASCII based on the identifier that correlates the ASCII and image data that is stored in temporary memory. Unlike Froessl, the claimed

embodiment attaches the property to a position in the document and the property preserves the original data used for insertion within the range. Accordingly, Saunders and Froessl fails to disclose or suggest the claimed third method.

Therefore, the Office Action fails to provide one or more references that reasonably teach or suggest the claimed text input processor interface.

(ii) Claims 23-26

The Office Action fails to provide one or more references that reasonably teach or suggest the claimed steps of "storing the original raw data in a property that is attachable to a document, specifying a range within the document, wherein the range utilizes at least one floating position, and attaching the property to the range specified within the document." The cited references, Saunders and Froessl, fail to teach or suggest, among other things, storing original raw data in a property that is attachable to a document and attaching the property to the range specified within the document. Accordingly, the Office has failed to meet its burden of establishing a *prima facie* case because all the claimed requirements are not suggested or taught by the cited references.
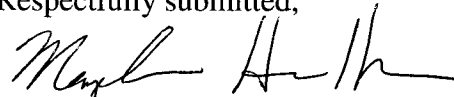
The Office Action seems to concede that Saunders does not teach or suggest storing original raw data in a property that is attachable to document.

However, the Office Action seems to contend, despite this deficiency, Saunders still discloses the claimed act of attaching the property to a range specified within the document. Appellants disagree because the claimed storing and attaching are tied by the claimed property that preserves the original raw data. Reading the claim as whole, if Saunders does not disclose storing the original data in the claimed property it follows that Saunders does not disclose the claimed attaching of the claimed property.

The Office Action seems to cite Froessl column 7, lines 63-column 8, lines 5 to provide the noted deficiencies. As indicated above Froessl does not teach or suggest the claimed property or attaching the property "within" the document. Instead Froessl column 8, line 20-25 associates image data and the ASCII based on the identifier that correlates the ASCII and image data that is stored in temporary memory. Unlike Froessl, the claimed embodiment attaches the property to a position in the document and the property preserves the original data. Accordingly, Saunders and Froessl fails to disclose or suggest the claimed storing and attaching acts.

Therefore, the Office Action fails to provide one or more references that reasonably teach or suggest the claimed storing and attaching acts.

Respectfully submitted,

Monplaisir Hamilton
Reg. No. 54,781

SHOOK, HARDY, & BACON L.L.P.
2555 Grand Blvd.
Kansas City, MO 64108-2613
Tel.:   816/474-6550
Fax: 816/421-5547

VIII. CLAIMS APPENDIX

1.   A machine-readable medium having instructions stored thereon for execution by a processor to implement a computer program providing a common text framework through which applications and handlers for input devices can interact, comprising:

a text store interface to permit an application having a document of primarily text to expose the document as an abstraction, the text store interface comprising,

a text stream interface in which the abstraction of the document appears as an array, a position within the document represented as an offset from a beginning of the array,

a dynamic text interface in which the abstraction of the document is such that a position within the document is represented as a floating anchor to a node, and

a text processor input method for attaching a property to the document in at least one position in the document, wherein the property preserves originally entered data in order to facilitate text correction; and

a text input processor interface to permit a handler for an input device to access the abstraction of the document and to insert additional text into the document.

2.   The medium of claim 1, wherein the text store interface comprises:

a method for selecting at least one of the text stream interface and the dynamic text interface by which to expose the document as the abstraction, wherein the method selects the text stream interface for documents stored as an array and the dynamic interface for documents stored in a tree-based structure.

3. The medium of claim 1, wherein the text input processor interface comprises a range object in which a range within the document is specified as two positions within the abstraction of the document, such that the handler inserts the additional text into the document and accesses the abstraction of the document at the range specified by the range object.

4. The medium of claim 3, wherein insertion is accomplished via a first method of the text input processor interface and the access is accomplished via a second method of the text processor interface.

5. The medium of claim 3, wherein the text input processor interface further permits the handler for the input device to attach the property to the document at the range specified by the range object.

20. A machine-readable medium having instructions stored thereon for execution by a processor to implement a computer program providing a common text framework through which applications and handlers for input devices can interact, comprising:

a text input processor interface to permit a handler for an input device to

access an abstraction of a document of mostly text of an application and to

insert additional text into the document, comprising:

a range object in which a range within the document is specified as two

positions within the abstraction of the document, such that the handler inserts

the additional text into the document and accesses the abstraction of the

document at the range specified by the range object;

a first method by which insertion is accomplished;

a second method by which access is accomplished; and,

a third method by which the handler is able to attach a property to the

document at the range specified by the range object, the property providing

access to original data used for insertion of text within the range.

21.   The medium of claim 20, wherein the framework further

comprises a text store interface to permit the application having the document of

primarily text to expose the document as an abstraction.

22.   The medium of claim 21, wherein the text store interface

comprises:

a text stream interface in which the abstraction of the document appears

as an array, a position within the document represented as an offset from a

beginning of the array; and,

a dynamic text interface in which the abstraction of the document is such

that a position within the document is represented as a floating anchor to a node,

such that the application selects at least one of the text stream interface and the dynamic text interface by which to expose the document as the abstraction.

23.   A method to preserve original raw data in a common text framework that provides interaction between an application and a plurality of input devices, the method comprising:

receiving the original raw data from at least one of the plurality of input devices;

storing the original raw data in a property that is attachable to a document;

specifying a range within the document, wherein the range utilizes at least one floating position; and

attaching the property to the range specified within the document.

24.   The method of claim 23, wherein at least two of the plurality of input devices can simultaneously provide the original raw data.

25.   The method of claim 23, further comprising:

providing a context that includes additional information about the original raw data and

utilizing the context to convert the received original raw data into text.

26.   The method of claim 25, wherein the context is determined by

the range specified within the document.

IX.   EVIDENCE APPENDIX

None.

X.   RELATED PROCEEDINGS APPENDIX

None.